

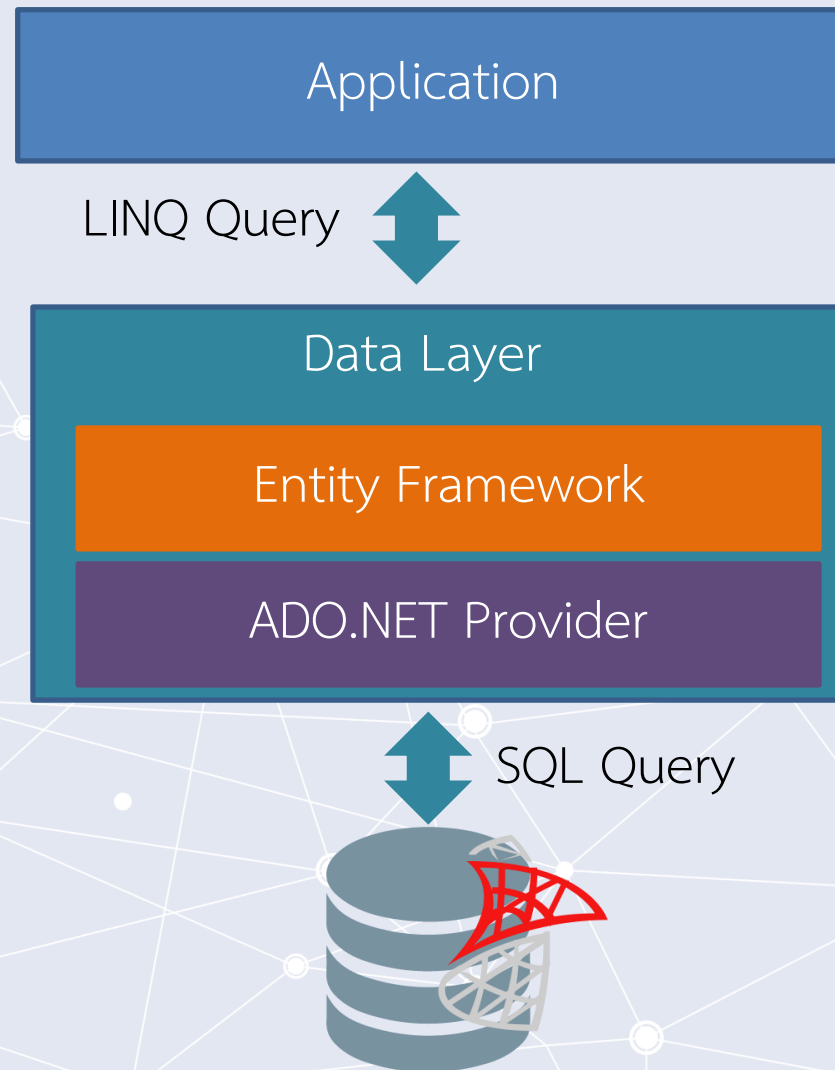
ระบบฐานข้อมูล

Chapter 05 : Entity Framework (EF): LINQ

อ.นุชนาถ สัตย์วินิจ

ภาควิชาวิศวกรรมคอมพิวเตอร์ วิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์

Entity Framework



Entity Framework

- เป็น Library ที่ทำหน้าที่จัดการฐานข้อมูล ร่วมกับการเขียน Class บนโปรแกรมที่เขียนขึ้น
- สร้าง Layer ทำหน้าที่เป็น Database Model ขึ้นมาเป็น Class ใน Project ซึ่งจะ Mapping ตัว Class กับ Table View และ Stored Procedure จากฐานข้อมูลมาไว้บน Project

Entity Framework

ADO.Net

```
String strConnString = "Server=localhost;UID=sa;PASSWORD=;database=mydatabase;Max Pool Size=400;Connect Timeout=600;";

SqlConnection objConn = new System.Data.SqlClient.SqlConnection(strConnString);
objConn.Open();

String strSQL = "SELECT * FROM customer";

SqlDataReader dtReader;
objCmd = new System.Data.SqlClient.SqlCommand(strSQL, objConn);
dtReader = objCmd.ExecuteReader();

objConn.Close();
objConn = null;
```

LINQ

```
using (var db = new myDatabaseEntities())
{
    var ds = db.CUSTOMER.ToList();
}
```

LinQ Syntax

- SELECT
- INSERT
- UPDATE
- DELETE

LinQ Syntax: SELECT

- การ SELECT ข้อมูลทั้งหมด
 - SQL => SELECT * FROM Physician
 - LINQ => var ds = (from p in db.Physicians select p).ToList();
- การ SELECT ข้อมูลไม่ซ้ำ
 - SQL => SELECT DISTINCT Position FROM Physician
 - LINQ => var ds = (from p in db.Physicians select p.Position).Distinct().ToList();

LinQ Syntax: SELECT

- การใช้ where
 - SQL => SELECT * FROM Physician WHERE <condition>
 - LINQ => var ds = (from p in db.Physicians whrer p.<condition> select p).ToList();
 - var ds = var ds = (from p in db.Physicians where p.Position == "Attending Physician" select p).ToList();

LinQ Syntax: SELECT

- การใช้ IN / NOT IN

- SQL => SELECT * FROM Physician WHERE Position in ('Attending Neonatologist', 'Attending Neurologist')

- LINQ =>

- ```
var strPos = (from p in db.Physicians where p.Position ==
"Attending Neonatologist" || p.Position == "Attending Neurologist"
select p.Position);
```

- ```
var ds = (from p in db.Physicians where  
strPos.Contains(p.Position) select p).ToList();
```


LinQ Syntax: SELECT

- การใช้ LIKE / NOT LIKE

- SQL => SELECT * FROM Physician WHERE Name LIKE '%a%'

- LINQ =>

- string name = "a";

- var ds = (from p in db.Physicians where p.Name.Contains(name)
select p).ToList();

LinQ Syntax: SELECT

- การใช้ ORDER BY
 - SQL => `SELECT * FROM Physician ORDER BY Name DESC`
 - LINQ => `var ds = (from p in db.Physicians orderby p.Name descending select p).ToList();`

LinQ Syntax: SELECT

- การใช้ ORDER BY
 - SQL => `SELECT * FROM Physician ORDER BY Position ASC, Name DESC`
 - LINQ => `var ds = (from p in db.Physicians orderby p.Position ascending, p.Name descending select p).ToList();`

LinQ Syntax: SELECT

- การใช้ BETWEEN

- SQL => SELECT * FROM Physician WHERE Name BETWEEN 'A' AND 'E'

- LINQ =>

- string a = "A";

- string b = "E";

- var ds = (from p in db.Physicians where p.Name.CompareTo(a) >= 0 && p.Name.CompareTo(b) <= 0 select p).ToList();

LinQ Syntax: SELECT

- การ Compare DateTime

- SQL => SELECT * FROM Appointment WHERE

- Appointment.StartA BETWEEN '2018-04-01 00:00:00' AND '2018-04-30 23:59:59'

- LINQ =>

- DateTime dtFrom = Convert.ToDateTime("2018-04-01");

- DateTime dtTo = Convert.ToDateTime("2018-04-30");

- var ds = (from p in db.Appointments where p.StartA >= dtFrom && p.StartA <= dtTo select p).ToList();

Questions & Answers